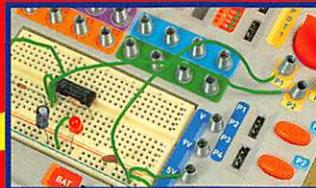
...e costruisci il tuo LABORATORIO DIGITALE

6,90 €

46

HARDWARE

DIGITALE DI BASE

MICROCONTROLLER

DIGITALE AVANZATO

TOTALMENTE PROGRAMMABILE!!!

9 771824 363008 41046

Peruzzo & C.

Direttore responsabile:
ALBERTO PERUZZO
Direttore Grandi Opere:
GIORGIO VERCELLINI
Consulenza tecnica e traduzioni:
CONSULCOMP S.n.c.
Pianificazione tecnica
LEONARDO PITTON

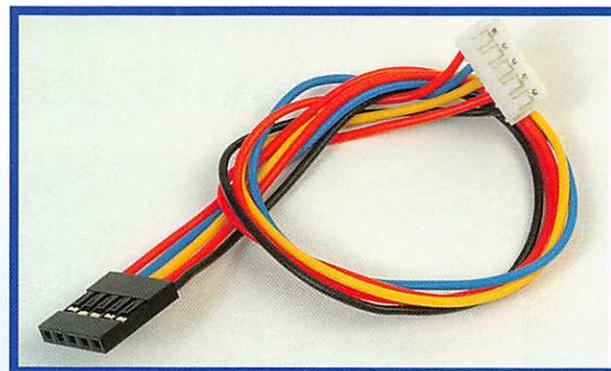
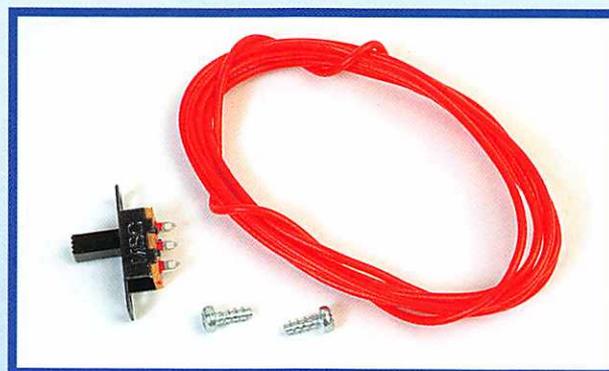
Direzione, Redazione, Amministrazione: viale Ercole Marelli 165, Tel. 02/242021, 20099 Sesto San Giovanni (Mi). Pubblicazione settimanale. Registrazione del Tribunale di Monza n. 1738 del 26/05/2004. Spedizione in abbonamento postale gr. II/70; autorizzazione delle Poste di Milano n. 163464 del 13/2/1963. Stampa: Grafiche Porpora s.r.l., Cernusco S/N (MI). Distribuzione SO.DI.P. S.p.A., Cinisello Balsamo (MI).

© 2004 F&G EDITORES, S.A.
© 2005 PERUZZO & C. s.r.l. Tutti i diritti sono riservati. Nessuna parte di questa pubblicazione può essere riprodotta, archiviata su sistema recuperabile o trasmessa, in ogni forma e con ogni mezzo, in mancanza di autorizzazione scritta della casa editrice. La casa editrice si riserva la facoltà di modificare il prezzo di copertina nel corso della pubblicazione, se costretta da mutate condizioni di mercato.

"ELETTRONICA DIGITALE"
si compone di
70 fascicoli settimanali
da suddividere
in 2 raccoglitori.

RICHIESTA DI NUMERI ARRETRATI. Per ulteriori informazioni, telefonare dal lunedì al venerdì ore 9.30-12.30 all'ufficio arretrati tel. 02/242021. Se vi mancano dei fascicoli o dei raccoglitori per completare l'opera, e non li trovate presso il vostro edicolante, potrete riceverli a domicilio rivolgendovi direttamente alla casa editrice. Basterà compilare e spedire un bollettino di conto corrente postale a PERUZZO & C. s.r.l., Ufficio Arretrati, viale Marelli 165, 20099 Sesto San Giovanni (MI). Il nostro numero di c/c postale è 42980201. L'importo da versare sarà pari al prezzo dei fascicoli o dei raccoglitori richiesti, più le spese di spedizione € 3,10 per pacco. Qualora il numero dei fascicoli o dei raccoglitori sia tale da superare il prezzo globale di € 25,82 e non superiore a € 51,65, l'invio avverrà per pacco assicurato e le spese di spedizione ammonteranno a € 6,20. La spesa sarà di € 9,81 da € 51,65 a € 103,29; di € 12,39 da € 103,29 a € 154,94; di € 14,98 da € 154,94 a € 206,58; di € 16,53 da € 206,58 in su. Attenzione: ai fascicoli arretrati, trascorse dodici settimane dalla loro distribuzione in edicola, viene applicato un sovrapprezzo di € 0,52, che andrà pertanto aggiunto all'importo da pagare. Non vengono effettuate spedizioni contrassegno. Gli arretrati di fascicoli e raccoglitori saranno disponibili per un anno dal completamento dell'opera. IMPORTANTE: è assolutamente necessario specificare sul bollettino di c/c postale, nello spazio riservato alla causale del versamento, il titolo dell'opera nonché il numero dei fascicoli e dei raccoglitori che volete ricevere.

IN REGALO in questo fascicolo

1 Cavetto a cinque fili con due connettori a cinque vie

IN REGALO nel prossimo fascicolo

1 Commutatore
1 Spezzone di filo flessibile rosso
2 Viti

COME RACCOGLIERE E SUDDIVIDERE L'OPERA NELLE 4 SEZIONI

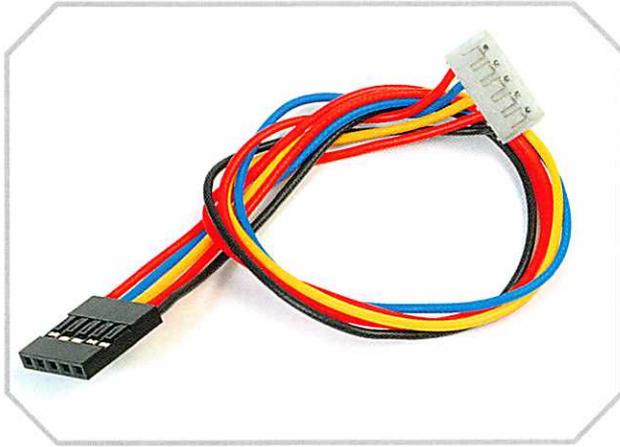
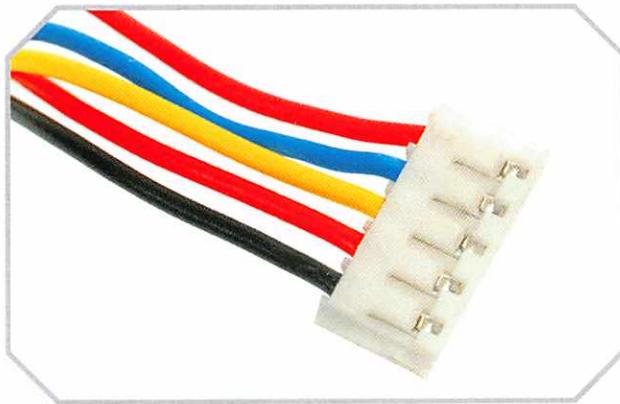
L'Opera è composta da 4 sezioni identificabili dalle fasce colorate, come indicato sotto. Le schede di ciascun fascicolo andranno suddivise nelle sezioni indicate e raccolte nell'apposito raccoglitore, che troverai presto in edicola. Per il momento, ti consigliamo di suddividere le sezioni in altrettante cartellette, in attesa di poterle collocare nel raccoglitore. A prima vista, alcuni numeri di pagina ti potranno sembrare ripetuti o sbagliati. Non è così: ciascuno fa parte di sezioni differenti e rispecchia l'ordine secondo cui raccogliere le schede.
Per eventuali domande di tipo tecnico scrivere al seguente indirizzo e-mail: elettronicadigitale@microrobots.it

Hardware Montaggio e prove del laboratorio

Digitale di base Esercizi con i circuiti digitali

Digitale avanzato Esercizi con i circuiti sequenziali

Microcontroller Esercizi con i microcontroller

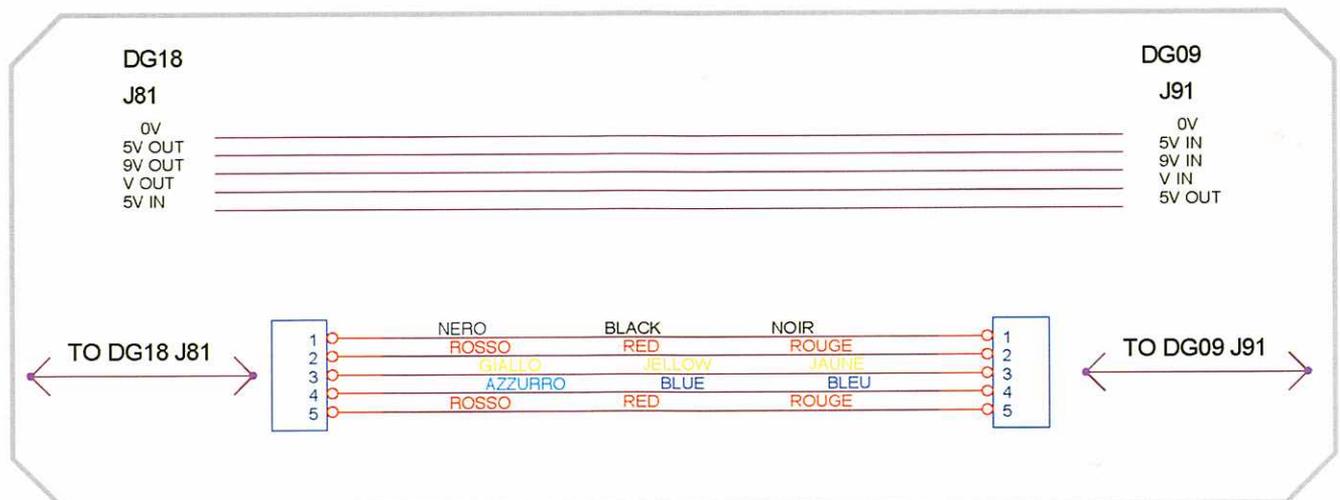
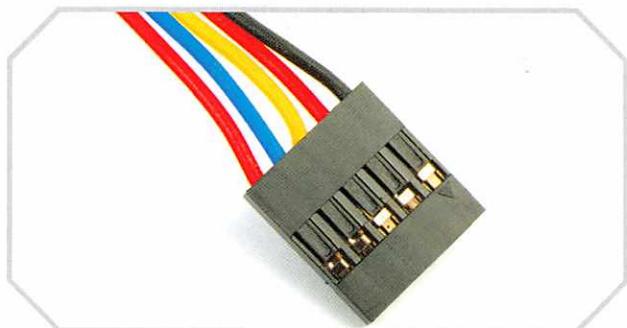
Trasferimento di alimentazione

Componenti forniti in questo fascicolo.

Dettaglio del connettore bianco.

Con questo fascicolo viene fornito il cavetto di collegamento dell'alimentazione fra i due pannelli che compongono il laboratorio. Si tratta di un cavetto a cinque fili terminato su due connettori.

Il cavetto dispone di due connettori a cinque vie diversi fra loro, uno di essi – di colore bianco – si collega in modo permanente al pannello principale; l'altro – di colore nero – anche se normalmente rimarrà sempre collegato può essere scollegato con facilità nel caso in cui, per qualche motivo, sia necessario separare i due pannelli del laboratorio.

Installazione

Per installare questo cavetto si inserisce il connettore di colore bianco attraverso il foro a forma ellittica situato in cima alla zona 1, molto vicina al display a sette segmenti del contatore. Dobbiamo far passare tutto il cavo, lasciandone fuori un pezzo di circa 5 centimetri.

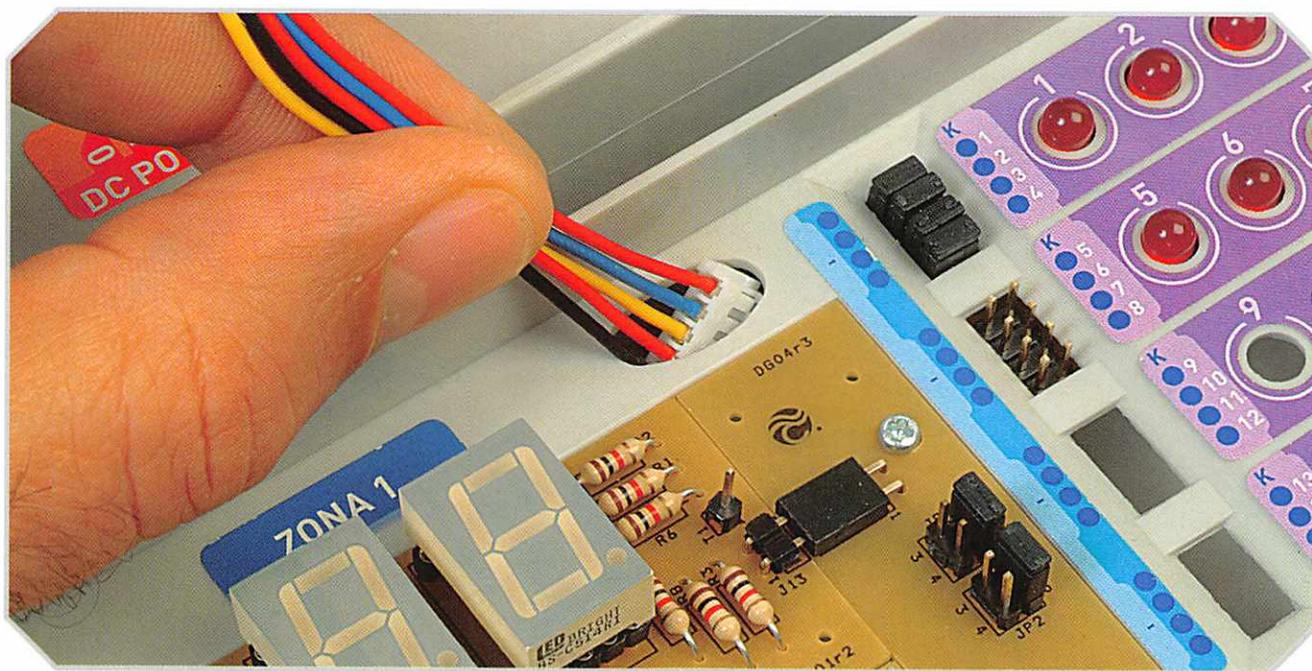
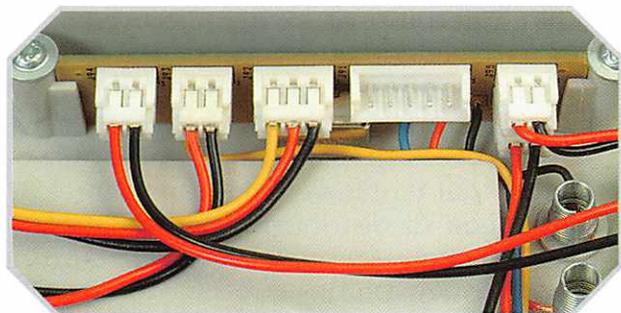
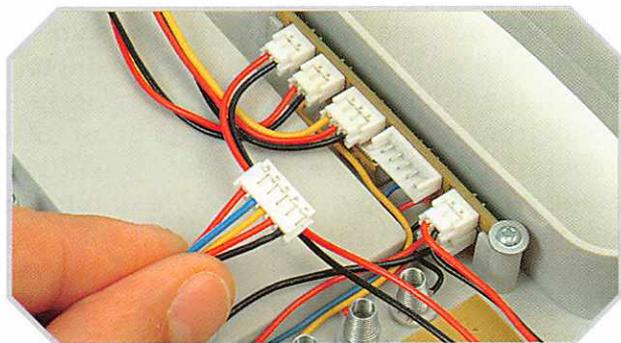
Il connettore bianco si collega su J91 della scheda di distribuzione dell'alimentazione DG09, in questo caso non c'è praticamente possibilità di errore, in quanto è rimasto solamente un unico connettore a cinque vie li-

Schema elettrico del cavetto e tensioni su ogni filo.

Dettaglio del connettore nero.

Tramite questo foro bisogna far passare il connettore bianco del cavetto.

bero e può essere collegato unicamente se orientato in modo corretto. Dopo aver messo uno di fronte all'altro il connettore del cavetto e della scheda, sarà sufficiente inserire il connettore del cavetto tenendo ferma la scheda per evitare di romperla. Il connettore maschio J91 dispone lateralmente di un dispositivo di ancoraggio che evita a entrambi i connettori di sganciarsi. Questa operazione corrisponde all'installazione fissa del cavetto.

Connettore nero

Il connettore nero del cavetto si unirà al momento opportuno al connettore J81 della scheda DG18 del prossimo fascicolo. Questa scheda fornisce da un lato le diverse tensioni di alimentazione da 0 V, 9 V e la tensione variabile che identifichiamo come V, ottenuta dall'alimentatore esterno; mentre dall'altro lato riceve l'alimentazione da 5 V della scheda di distribuzione dell'alimentazione, situata nel pannello inferiore, per distribuirla al resto dei circuiti installati sul pannello superiore.

Nelle etichette del pannello superiore si

Il cavetto deve passare lasciando fuori un pezzo di circa 5 cm.

Il cavetto si collega su questo connettore.

Vista interna prima del collegamento del cavetto.

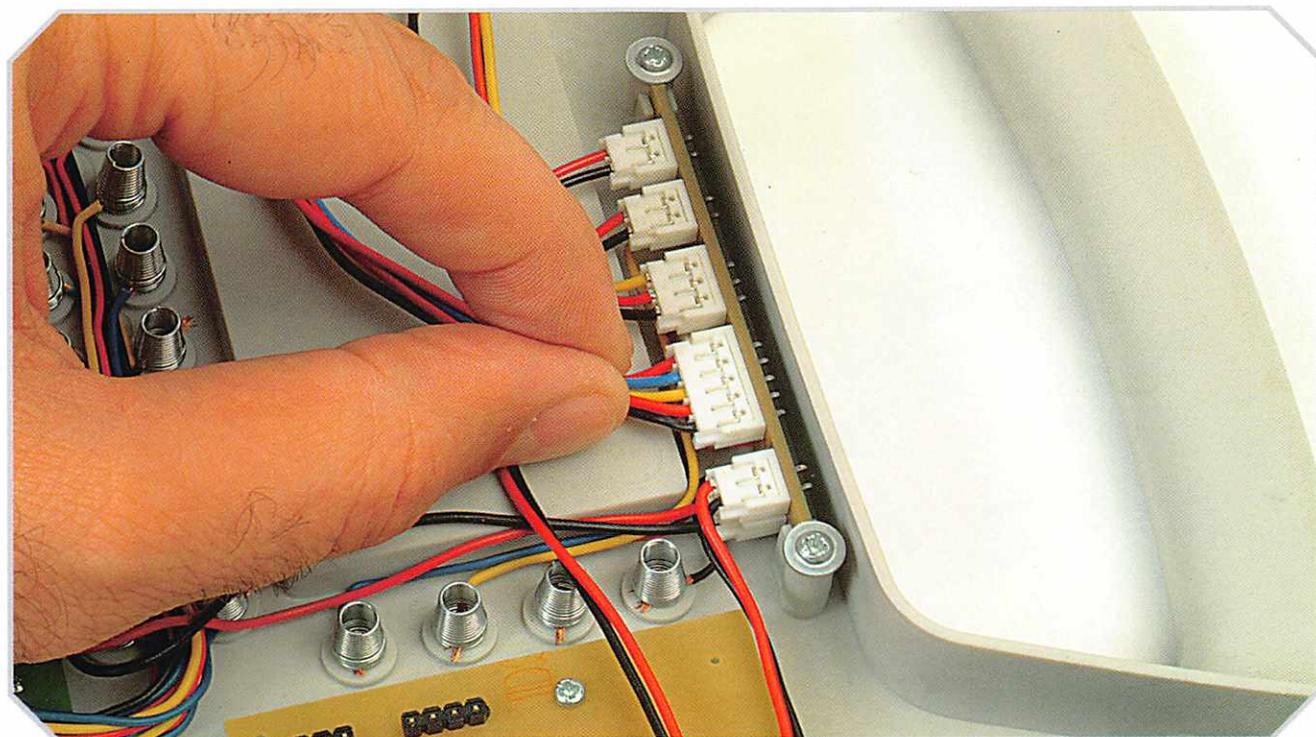
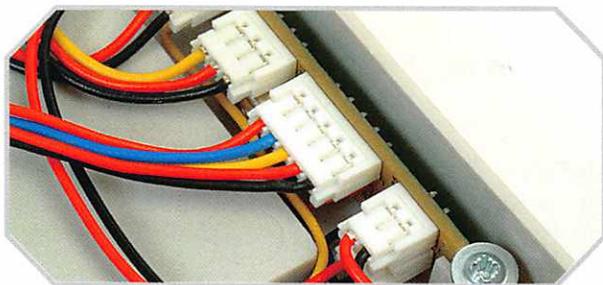
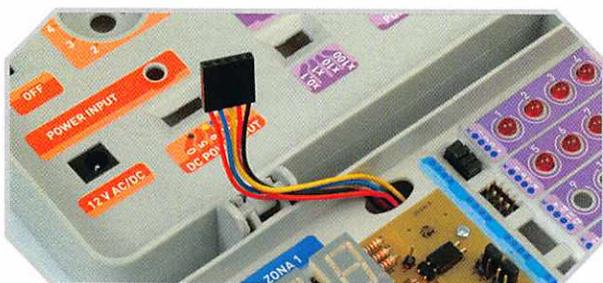
identifica il connettore J81 della scheda DG18 come DCPOWER OUT, su questa etichetta troviamo anche una serie di piccoli pallini colorati riferiti al colore di ogni conduttore di questo cavetto per evitare errori di collegamento. I terminali di questo connettore maschio sono accessibili dall'esterno del pannello superiore.

Segnali

Descriviamo ora il compito di ogni filo di questo cavetto.

Il filo 1, di colore nero, unisce fra loro i terminali 1 dei due connettori, è il riferimento dell'alimentazione, ovvero 0 V, che è anche il negativo dell'alimentazione.

Il filo 2, di colore rosso, porta l'alimentazione da 5 V, che fornisce il terminale 2 della scheda DG18, fino al terminale 2 del connettore J91 della scheda DG09.

Il filo 3, di colore giallo, porta l'alimentazione da 9 V dal terminale 3 del connettore J81 della scheda DG18, fino al terminale 3 del connettore J91 della scheda DG09.

Collegamento a J91 di DG09.

Il connettore rimane agganciato sul suo corrispondente della scheda.

Ecco come deve rimanere questo connettore, sino a quando non avremo a disposizione la scheda DG18.

Il filo 4, di colore azzurro, porta l'alimentazione variabile V dal terminale 4 del connettore J81 della scheda DG18 fino al terminale 4 del connettore J91 della scheda DG09.

Il filo 5, di colore rosso, porta l'alimentazione da 5 V dal terminale 5 del connettore J91 della scheda di distribuzione dell'alimentazione DG09 fino al terminale 5 del connettore J81 della scheda DG18.

Precauzioni

Questo cavetto, quando si collega il primo blocco di batterie, è alimentato, ma solo sui terminali 0 V e 5 V, ovvero agli estremi, sui terminali 0 e 5 dello stesso. Quindi, anche se il connettore è femmina, il che evita quasi tutti i contatti accidentali, è necessario fare attenzione a questo lato del connettore; inoltre la sua posizione è piuttosto vicina alla zona più utilizzata del laboratorio.

Presto vi verrà fornita la scheda DG18 che risolverà il problema. Su questo cavetto non saranno disponibili i 9 V delle batterie.

Vista generale del laboratorio.

Influenza della temperatura

In questo esercizio faremo delle prove con la temperatura e verificheremo la sua influenza sul funzionamento del circuito.

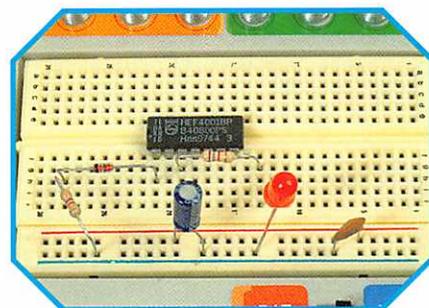
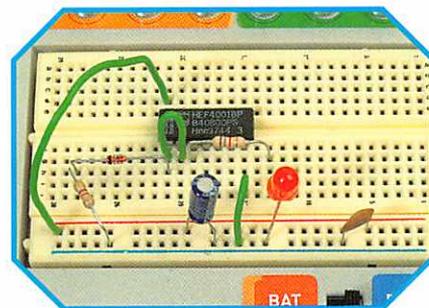
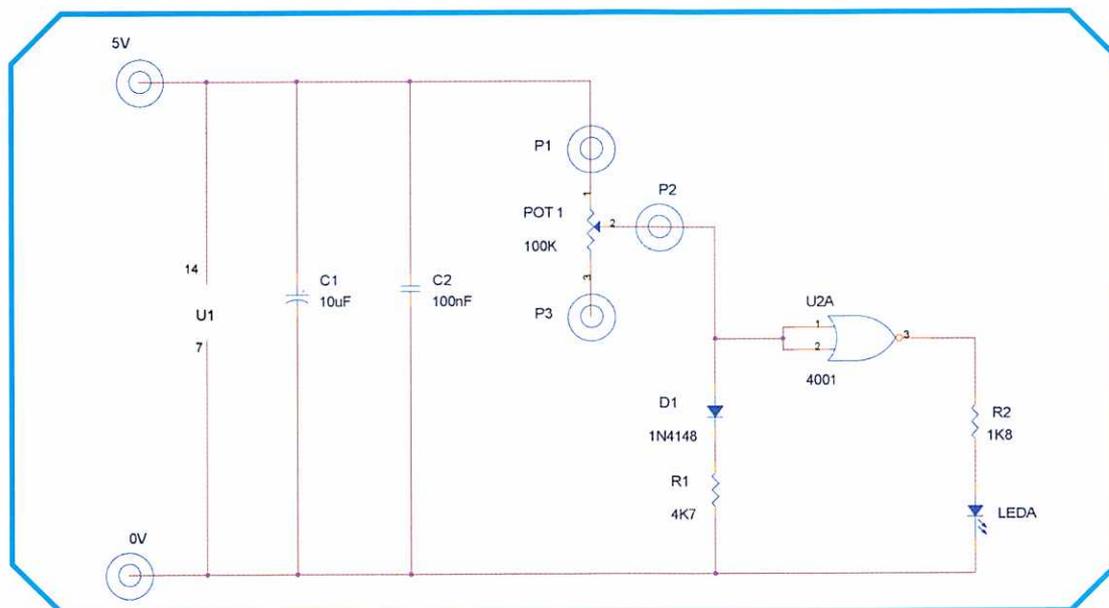
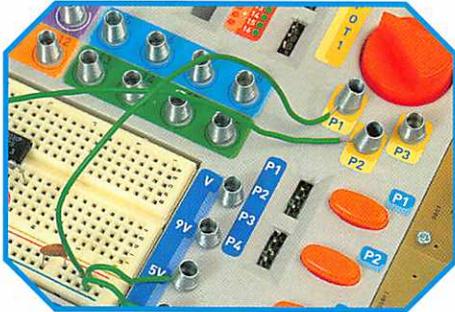
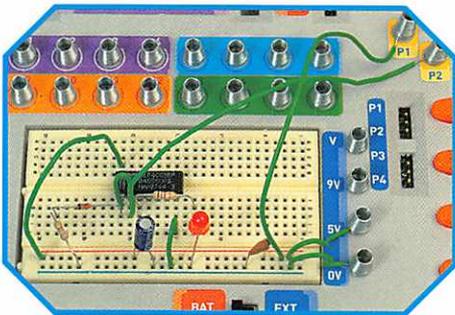
L'idea

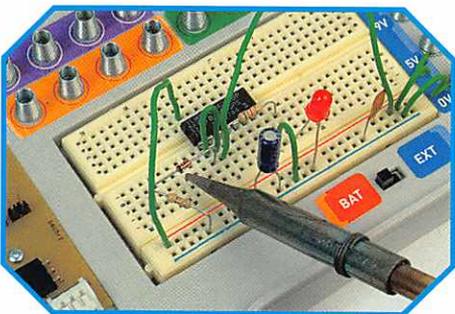
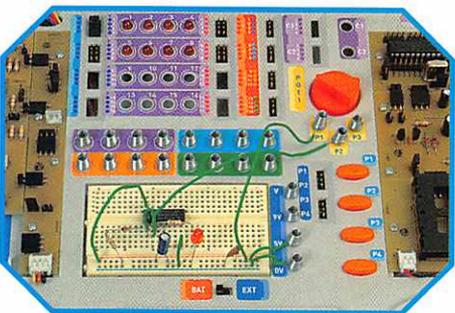
La temperatura può influenzare in modo molto importante il funzionamento dei circuiti, e non solo perché può arrivare a distruggerli o apportare avarie, ma anche perché può interferire sul funzionamento interno del circuito stesso.

Questo problema si apprezza durante il progetto, tuttavia quando si lavora con dispositivi già costruiti non è molto evidente, in quanto il progettista ha già risolto in precedenza i problemi. I circuiti integrati dispongono di una compensazione interna della temperatura, quindi l'influenza di questi ultimi sugli stessi è piuttosto bassa, sempre che si lavori all'interno dei limiti che ne garantiscono il funzionamento e osservando i parametri specificati.

Scheda Bread Board con i componenti utilizzati.

Cablaggio interno della scheda.

Il circuito

L'elemento fondamentale di questo esperimento è un diodo 1N4148 di uso comune, il resto del circuito si utilizza per evidenziare la variazione di tensione originata da una differenza di temperatura. Se guardiamo lo schema del circuito potremo vedere che il diodo D1, del tipo 1N4148, è collegato fra il positivo e il negativo dell'alimentazione, separato dal negativo da una resistenza R1 da 4K7 e dal positivo da un'altra resistenza, che corrisponde alla resistenza fra i terminali P1 e P2 del potenziometro POT1.

Schema del circuito di prova.

Collegamenti di POT1.

L'alimentazione può essere da 5 oppure da 9 V.

Regoleremo il potenziometro fino a quando si spegnerà il LED.

Con il saldatore si ottiene un forte aumento di temperatura.

Esperimento completato.

L'anodo del diodo è collegato all'ingresso di una porta invertente costruita con una delle porte NOR del circuito integrato 4001.

In generale quando l'ingresso della porta è a livello logico 1, la sua uscita è a livello logico 0 e il LED A non è illuminato, ma quando l'ingresso è a livello basso la sua uscita passa a livello alto e il LED si illumina. Ruotando il potenziometro verso un lato o l'altro otterremo l'illuminazione o lo spegnimento del LED. C1 e C2 sono condensatori di filtro per l'alimentazione.

Montaggio

Il montaggio si esegue come d'abitudine, facendo particolare attenzione alla polarità del diodo D1, del LED A e del condensatore elettrolitico C1, così come per i LED e i condensatori elettrolitici rimanenti. Il circuito può essere alimentato indistintamente a 9 o 5 V.

L'esperimento

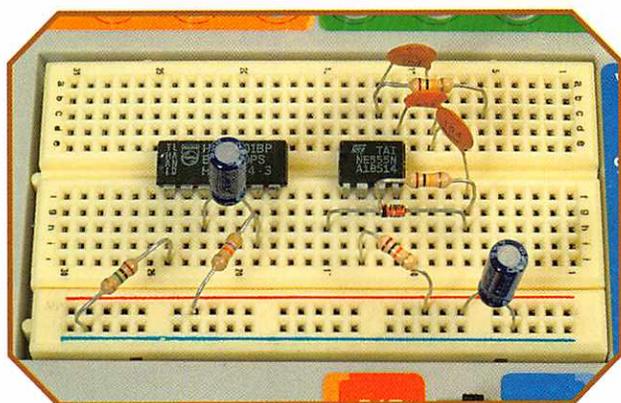
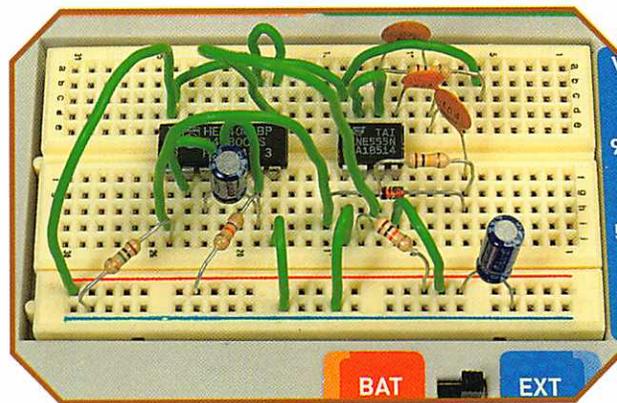
La caduta di tensione in un diodo diminuisce di 0,2 millivolt per ogni grado in più di temperatura.

Dopo aver verificato il funzionamento del circuito, realizziamo l'esperimento. Azionando il comando del potenziometro lo si ruota lentamente sino a quando si spegnerà il LED. Quando riscaldiamo il diodo, cosa che possiamo fare con la mano, il LED inizia a illuminarsi, ma se non otterremo un innalzamento sufficiente di temperatura, potremo avvicinare la punta del saldatore caldo al contenitore del diodo e vedremo illuminarsi il diodo LED. È necessario fare molta attenzione per non danneggiare la scheda Bread Board, o qualsiasi pezzo in materiale plastico del laboratorio. Possiamo raffreddare il diodo toccandolo con una superficie fredda, o mettendolo nel congelatore e regolando rapidamente POT1, quindi osserveremo che il diodo si illumina quando torna a temperatura ambiente.

LISTA DEI COMPONENTI

U1	Circuito integrato 4001
D1	Diodo 1N4148
R1	Resistenza 4K7, 5% 1/4 W (giallo, viola, rosso)
R2	Resistenza 1K8, 5% 1/4 W (marrone, grigio, rosso)
C1	Condensatore 10 µF elettrolitico
C2	Condensatore 100 nF
LEDA	Diodo LED rosso

Contatore bidirezionale

In questo circuito si controlla un contatore con tre pulsanti: uno produce il conteggio, l'altro provoca la diminuzione e il terzo lo azzera completamente.

Componenti sulla scheda Bread Board.

Cablaggio interno della scheda.

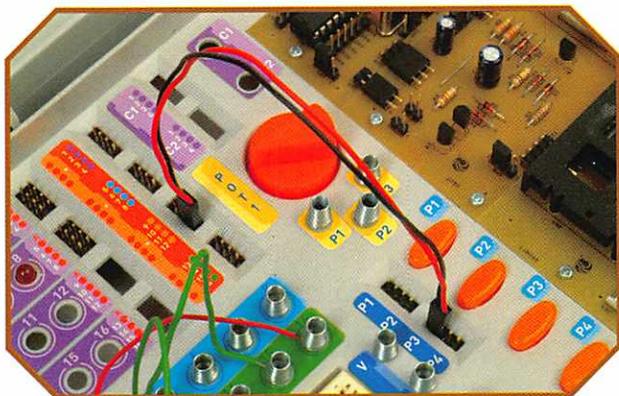
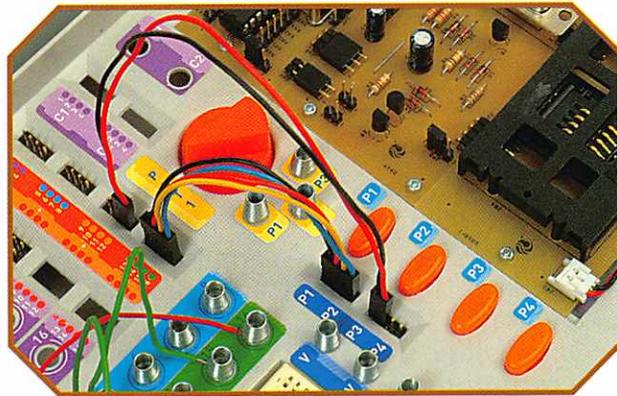
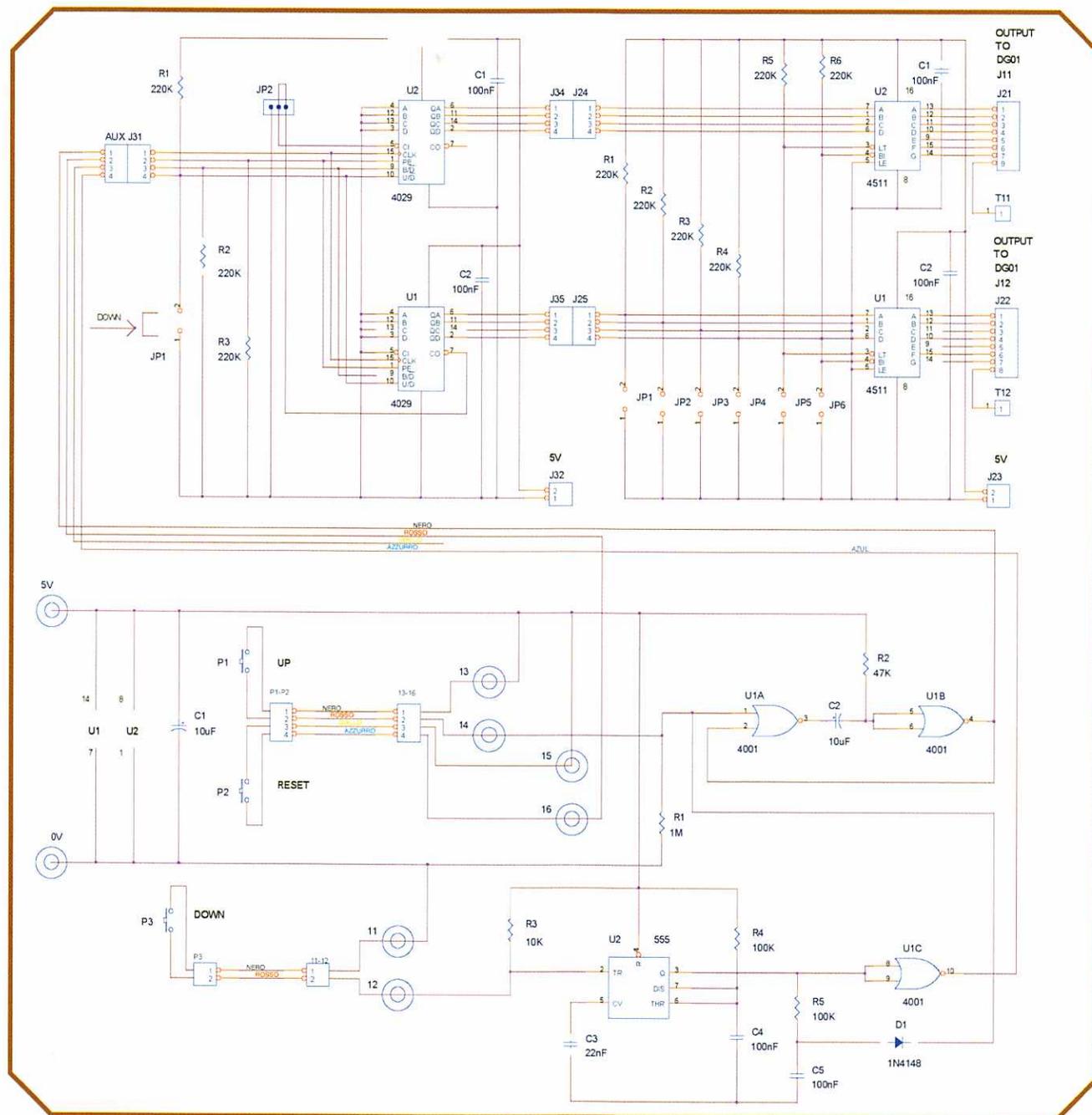
Scopo del circuito

Vogliamo un circuito che funzioni nel seguente modo: ogni volta che si preme un pulsante il contatore avanzerà di una unità mentre se ne azioniamo un altro retrocederà di una unità, tutto questo in modo automatico e senza bisogno di dover cambiare i ponticelli o i collegamenti; inoltre un terzo pulsante azzererà il conteggio.

Inizieremo il progetto seguendo un metodo leggermente diverso da quanto fatto finora. Il pulsante P1 azionerà un monostabile la cui uscita applicherà un unico impulso all'ingresso di clock del contatore. Per il pulsante P3 sarà necessario utilizzare un circuito addizionale per impostare a livello basso il terminale 2 del connettore J31 della scheda DG03, prima che l'impulso di clock arrivi all'ingresso di clock del contatore; in questo modo il contatore conterà a ritroso.

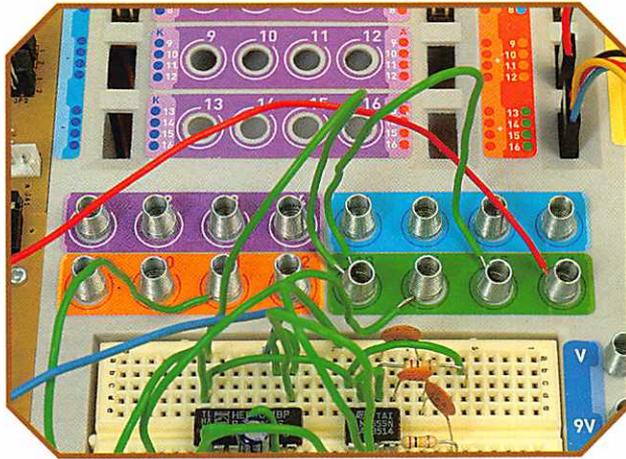
Vediamo ora, con l'aiuto dello schema, come si realizza il circuito.

Schema

Per descrivere lo schema inizieremo dai pulsanti e ne seguiremo il collegamento. Il pul-

Collegamenti del pulsante P3.

Collegamenti dei pulsanti P1 e P2.

Schema elettrico del contatore.

sante P1 attiva il monostabile formato dalle porte U1A e U1B del circuito integrato 4001. Ogni volta che si preme si applica un impulso al terminale 1 di J31 e il contatore avanza di una unità.

Ogni volta che si aziona P2 si imposta a zero il contatore.

Analizziamo ora il funzionamento del pulsante P3. Quando si aziona questo pulsante si attiva il monostabile formato dal circuito integrato U2, che è un 555 configurato come monostabile. L'uscita di questo circuito, che in condizioni di riposo è a livello basso, passa a livello alto e tramite una porta invertente arriva al terminale 4 di J31; quando questo terminale è a livello basso, se il contatore ri-

I collegamenti dei pulsanti sono disponibili sulle molle dalla 11 alla 16.

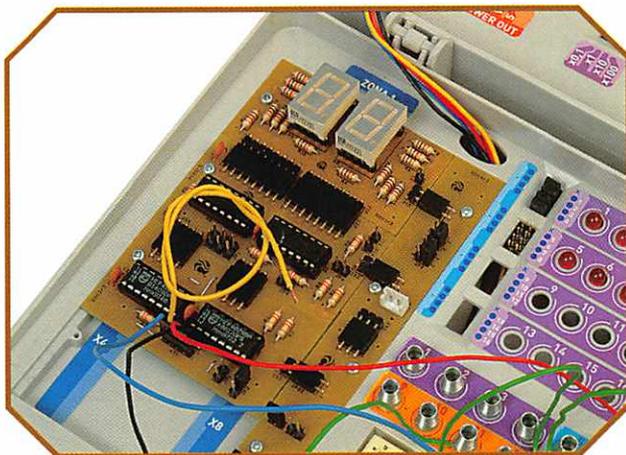
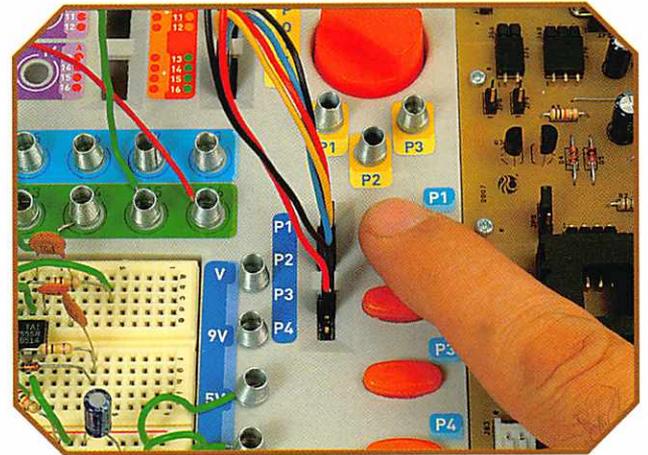
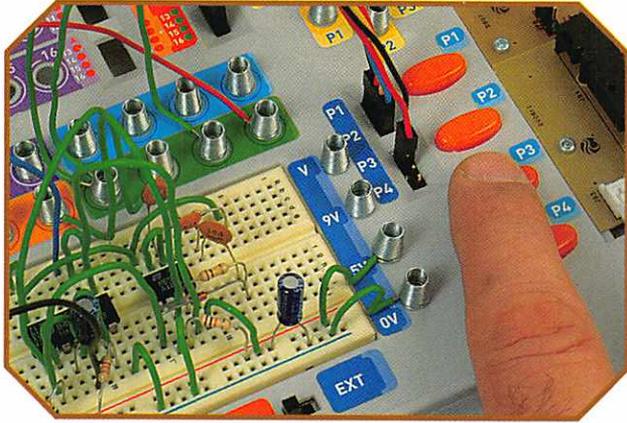
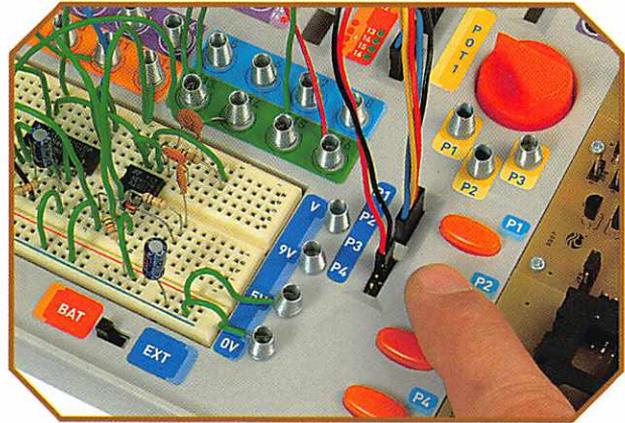
Circuito pronto per la prova.

ceve un impulso di clock il conteggio si riduce. Ogni impulso del monostabile che inverte il senso del conteggio, terminale 3 del 555, si trasmette all'ingresso del monostabile che applica impulsi di clock, cioè arriva al terminale 1 di U1A, ma con il ritardo provocato dalla rete di ritardo RC – formata dal condensatore C4 e dalla resistenza R4 – in modo che l'impulso che inverte il senso del conteggio arrivi prima al contatore dell'impulso di clock.

Dopo un determinato periodo di tempo il contatore torna allo stato di riposo: se si preme P1 conta in avanti, se si preme P3 conta indietro. La resistenza R1 scarica il condensatore C5 dopo aver prodotto l'attivazione, e il diodo D1 evita che gli impulsi di attivazione generati da P1 possano arrivare all'ingresso della porta U1C.

Montaggio

Il montaggio si realizza come d'abitudine, iniziando con l'inserimento dei componenti sulla scheda Bread Board, tenendo presente l'orientamento dei circuiti integrati e la polarità del diodo D1 e del condensatore elettrolitico C2.

Il pulsante P3 (DOWN) necessita solamente di due connessioni e si collega con il cavetto a due fili. I pulsanti P1 (UP) e P2 (RESET) si collegano mediante un cavetto a quattro fili. Il ponticello JP2 della scheda DG03 deve essere montato in modo che il contatore conti le de-

Contatore predisposto per l'esperimento.

Ogni volta che si preme P1 il contatore avanza.

Premendo P3 il conteggio diminuisce di una unità.

Il pulsante P2 si utilizza per azzerare il conteggio.

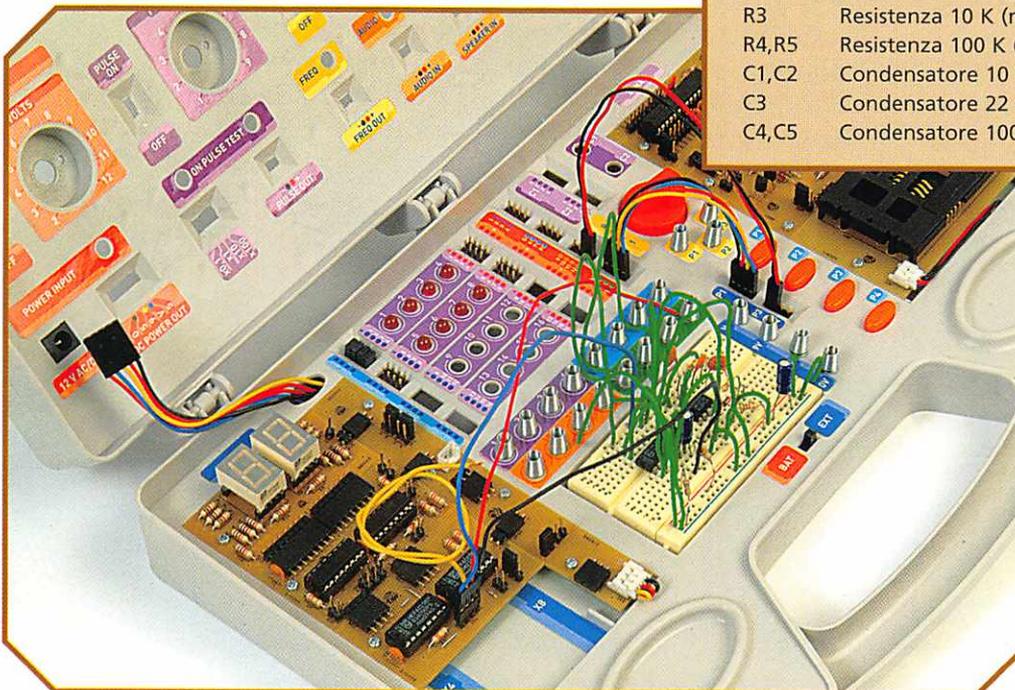
cine. I collegamenti al terminale J31 del contatore si eseguono con un cavetto terminato da un lato su un connettore, e dall'altro lato con i fili liberi spelati in punta.

Questo circuito si può alimentare indistintamente a 5 oppure a 9 V, verificando che le schede del contatore siano anch'esse alimentate alla tensione scelta.

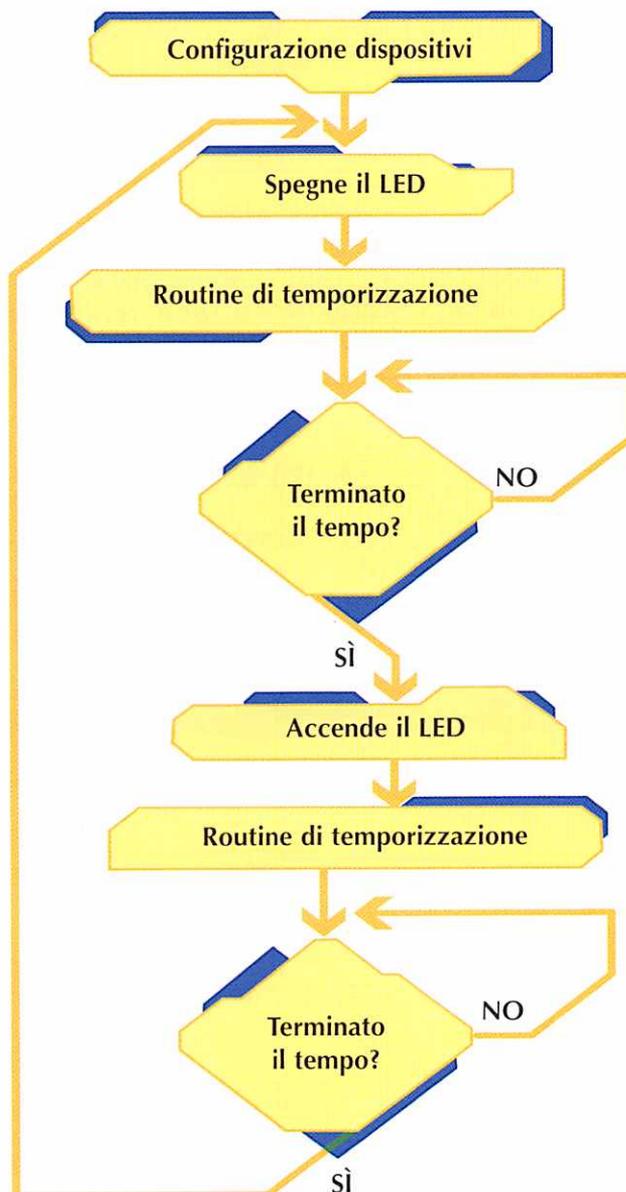
Funzionamento

Dopo aver verificato il lavoro svolto, si collega l'alimentazione e il circuito risulta pronto per il funzionamento. Se premiamo P1 il contatore avanza di una unità. Premendo varie volte il contatore avanzerà di diverse unità. Se successivamente premeremo P3 il contatore avanzerà in senso inverso, e se premeremo P2 il contatore si resetterà.

LISTA DEI COMPONENTI

U1	Circuito integrato 4001
U2	Circuito integrato 555
D1	Diodo 1N4148
R1	Resistenza 1 M (marrone, nero, verde)
R2	Resistenza 47 K (giallo, viola, arancio)
R3	Resistenza 10 K (marrone, nero, arancio)
R4,R5	Resistenza 100 K (marrone, nero, giallo)
C1,C2	Condensatore 10 µF
C3	Condensatore 22 nF
C4,C5	Condensatore 100 nF

Lista del laboratorio con l'esperimento completato.

Esercizi di ripasso

Lavoreremo con i nuovi dispositivi del microcontroller che abbiamo studiato, quindi faremo una serie di esercizi sul laboratorio che simuleranno problemi reali che potremmo trovare nel mondo della programmazione. Prima di iniziare a lavorare con questi dispositivi faremo alcuni esercizi che miglioreranno le nostre conoscenze e ci prepareranno per risolvere gli enunciati più complessi.

Organigramma corrispondente all'enunciato 1.

Enunciato 1: TMR0

Il primo esercizio si basa sulla funzione del Timer 0 (TMR0). Si tratta di sviluppare un programma che faccia lampeggiare un LED collegato a RB2 ogni 65,28 ms, utilizzando un range del divisore di 1:256 e preveda l'utilizzo del TMR0.

Per lo sviluppo di questo programma dobbiamo disporre unicamente di un'uscita per collegare il diodo LED e lavorare con il Timer 0. Nell'organigramma della figura si può vedere la semplicità del nostro obiettivo. Configureremo i dispositivi, che in questo caso sono solamente la porta B per l'uscita del LED e il predivisore di frequenza per il temporizzatore. Inizieremo poi il programma spegnendo il LED e chiamando la routine di temporizzazione, che ritornerà al programma principale dopo che sarà trascorso il tempo desiderato, accenderemo il LED e torneremo a utilizzare la routine di temporizzazione. Ripeteremo il ciclo spegnendo nuovamente il LED e ripetendo i passaggi precedenti.

Codice che risolve l'enunciato 1

Per sviluppare il codice dobbiamo anzitutto calcolare il predivisore di frequenza che vogliamo utilizzare per ottenere il tempo desiderato, e a questo scopo utilizzeremo la for-

Tempo = 4·(1/Fosc)Valore·Predivisore

Quindi:

65,28 ms = 4·(1/4$10^6$) Valore 256 —> Valore = 255

Formula per calcolare il tempo del temporizzatore.

```
;                        ESERCIZIO 1: TMR0
;                     ----------------------

;Gestione del timer 0 (TMR0) con il PIC16F870.

;Programma che fa lampeggiare il LED collegato a RB2 ogni 65,28 ms utilizzando
;il TMR0 con un range del predivisore di 1:256

;Il calcolo del valore da contare è:
;65,28ms=4*(1/4MHz)*valore*256 ---> valore=255

        LIST    P=16F870        ;Definiamo il nostro PIC
        INCLUDE "P16F870.INC"   ;File dei registri interni

        ORG     0
        GOTO    INIZIO
        ORG     5

;Programma principale.

INIZIO  clrf    PORTB
        bsf     STATUS,RP0      ;Passiamo al banco 1
        clrf    TRISB           ;Porta B uscite
        movlw   b'11010111'     ;Predivisore 256
        movwf   OPTION_REG
        bcf     STATUS,RP0      ;Torniamo al banco 0

LAMPEG  bcf     PORTB,2         ;Spegnamo il terzo LED (RB2)
        call    RITARDO         ;Saltiamo alla routine di ritardo
        bsf     PORTB,2         ;Accendiamo il terzo LED
        call    RITARDO
        goto    LAMPEG

RITARDO movlw   b'00000001'     ;Valore da contare 255
        movwf   TMR0            ;(complemento a 2)
        btfss   INTCON,T0IF
        goto    $-1             ;Torna all'istruzione precedente
        bcf     INTCON,T0IF
        return

        END
```

Codice che risolve l'enunciato 1.

mula della figura sottostante. In seguito intesteremo il nostro programma con i commenti pertinenti e le direttive appropriate in cui vengono definiti il microprocessore e le librerie da utilizzare. Configureremo la porta B come uscita e caricheremo sul registro OPTION_REG il valore del predivisore di frequenza. Il programma principale sarà un ciclo in cui si cancella il bit corrispondente al terminale RB2, si chiama la temporizzazione, si attiva nuovamente il terminale e si ritorna alla temporizzazione. Nella routine di temporizzazione caricheremo solamente il valore che abbiamo calcolato, e attenderemo che si compia la condizione che indica che il tempo è trascorso per ritornare al programma principale.

Organigramma corrispondente all'enunciato 2.

Enunciato 2: TMR0 e interrupt

In questo secondo esercizio combineremo l'utilizzo del TMR0 con quello degli interrupt, provocando un interrupt per overflow del TMR0. Si tratta di sviluppare un programma

```
;                        ESERCIZIO 2: Interrupt e TMR0
;
;Interrupt per overflow del TMR0 con il PIC16F870.

;Programma che fa lampeggiare il LED collegato a RB2 ogni 1,3 s mediante
;interrupt del TMR0 ogni 65,28 ms facendolo illuminare o spegnere solamente
;ogni 20 interrupt, utilizzando un range del predivisore di 1:256.

;Il calcolo del valore da contare è: 65,28ms=4*(1/4MHz)*valore*256 --> valore=255

        LIST    P=16F870        ;Definiamo il nostro PIC
        INCLUDE "P16F870.INC"   ;File dei registri interni

Cont1   equ     0x20

        ORG     0
        GOTO    INIZIO
        ORG     4
        GOTO    INT
        ORG     5
```

Codice nell'intestazione dell'enunciato 2.

```
;Programma principale.

INIZIO  clrf    PORTB
        bsf     STATUS,RP0      ;Passiamo al banco 1
        clrf    TRISB           ;Porta B uscite
        movlw   b'11010111'
        movwf   OPTION_REG
        bcf     STATUS,RP0
        movlw   b'10100000'     ;Attiviamo gli interrupt
        movwf   INTCON          ;per il TMR0
        movlw   b'00000001'
        movwf   TMR0            ;Inizializziamo il conteggio
        movlw   .20
        movwf   Cont1

NULLA   clrwdt
        goto    NULLA           ;Ciclo che non fa nulla sino
                                ;a quando non si genera un interrupt.
        END
```

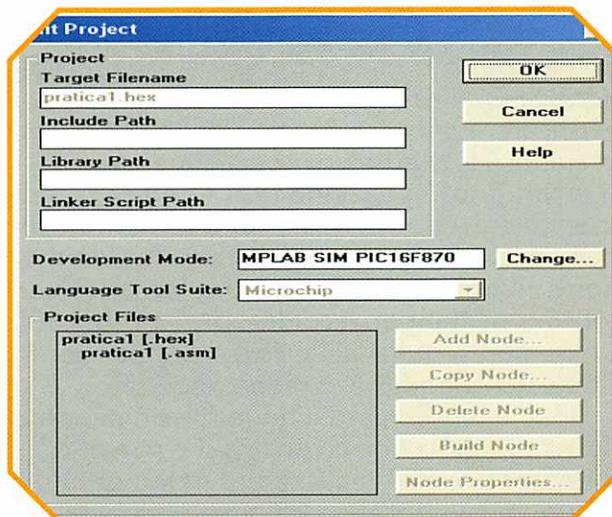
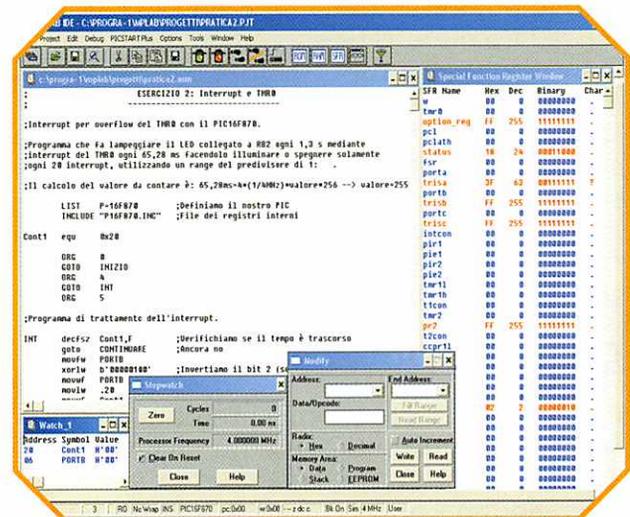
Programma principale.

Programma di trattamento dell'interrupt.

Torniamo a lavorare con MPLAB.

che faccia lampeggiare un LED collegato a RB2 ogni 1,3 s. A questo scopo utilizzeremo l'interrupt del TMR0 ogni 65,28 ms, in modo che si illumini o si spenga solamente dopo 20 interrupt, utilizzando un range del divisore di 1:256.

Questo esercizio utilizza gli interrupt, oltre ai dispositivi dell'esercizio precedente, quindi il modo con cui abbiamo previsto di risolverlo è molto diverso dal precedente. Nel programma principale si configurano i dispositivi: porta B, predivisore di frequenza e interrupt. In seguito inizializzeremo i valori del conteggio e in ultimo, entreremo nel ciclo in cui attenderemo che si generino gli interrupt.

Nel programma per il trattamento degli interrupt risiede la "difficoltà" dell'esercizio. La prima cosa da fare è verificare che sia trascorso il tempo totale desiderato, questo avviene quando si producono 20 interrupt del temporizzatore. Nel caso in cui la condizione non sia soddisfatta, ridurremo il contatore e attenderemo un nuovo interrupt; se sono stati generati 20 interrupt cambieremo lo stato dell'uscita e reinizializzeremo i valori del conteggio. Gli organigrammi presentati rispondono al funzionamento spiegato.

Per compilare i vostri codici dovrete utilizzare MPLAB e creare un progetto.

Videata tipica di simulazione con MPLAB.

Risultato della compilazione.

Codice che risolve l'enunciato 2

Dato che in questo esercizio dobbiamo utilizzare una variabile come contatore per i 20 interrupt, definiremo quest'ultima all'inizio del programma, riservando a essa un indirizzo nella memoria. Nelle direttive in cui definiamo dove si trova il codice del programma, dobbiamo inserire la routine di interrupt (ORG 4).

Nelle immagini della pagina precedente possiamo vedere come rimane l'intestazione del programma. Avendo ben chiare queste particolarità, lo sviluppo del codice non risulta molto complicato. Nel programma principale si configurano i dispositivi da utilizzare e se ne definiscono i valori, sia del predivisore che del registro di configurazione degli interrupt, per poter così abilitare ciò che ci interessa, ovvero il TMR0. Inizializzeremo i valori di conteggio e attenderemo in un ciclo infinito senza fare nulla. Nel programma per il trattamento degli interrupt si contano questi ultimi e se si arriva a 20, si cambia il valore dell'uscita mediante l'istruzione XOR, reinizializzando successivamente i valori del conteggio. Se non si è ancora arrivati a 20 caricheremo nuovamente il temporizzatore con il suo valore di conteggio e attenderemo che finisca e provochi un nuovo interrupt.

Compilazione

Per compilare i programmi presentati o quelli fatti da voi stessi utilizzeremo MPLAB. Ricordate che è necessario creare un progetto, ad esempio con il nome di pratica1, a cui aggiungere in seguito il codice in assembler associato. Aprite il file che avete associato al progetto per poterlo visualizzare e selezionate Build All dal menù Project o premete Ctrl+F10. In questo modo verrà compilato il codice e potrete verificare se è stato commesso qualche errore durante il confezionamento del programma. Eseguite questi passi con i due programmi realizzati e verificate che non contengano errori. Nel caso in cui la compilazione non avvenga con successo, tenendo aperto il codice sul display, fate un doppio click con il mouse sulla linea che indica l'errore nella compilazione, e automaticamente il software si posizionerà sulla linea di codice che contiene l'errore.

Quando la compilazione viene eseguita con successo si creano i file in codice macchina (.hex) che successivamente si scriveranno sul microcontroller.

Simulazione

Come abbiamo già visto negli esercizi precedenti la simulazione risulta molto comoda per verificare il corretto funzionamento dei programmi. Il fatto che un programma non generi errori di compilazione non significa che risponda correttamente ai requisiti del funzionamento. I due programmi presentati non sono facili da simulare, dato che lavorano con routine di temporizzazione o con interrupt. Dovremo eseguire delle forzature per verificare che rispondano in modo adeguato.

Dopo aver simulato il buon funzionamento del programma non ci rimarrà che scriverlo sul microcontroller e realizzare il montaggio del circuito per verificare che il funzionamento sia realmente quello desiderato.